Automated Log-Scale Quantization for Low-Cost Deep Neural Networks Supplementary Material

Sangyun Oh[1], Hyeonuk Sim[2], Sugil Lee[1], Jongeun Lee[1†]
[1]Department of Electrical Engineering, UNIST, Ulsan, Korea
[2]Department of Computer Science and Engineering, UNIST, Ulsan, Korea
{syoh,detective,sglee17,jlee}@unist.ac.kr

In this supplementary material, we attached some of the training propensity data recorded at the time of the paper experiments, along with the description, to assist the reader in understanding the effect of our training method.

1. The Effect of Select Tensor

Our proposed method optimizes model performance for STLQ with predefined *select ratio* in training algorithm. Therefore, we present some of the resulting trends in Figure 1 to help understand the effects of our method. We recorded this data when training ResNet-18 model [5] in Table (4) of the paper with the selection ratio of 15% (W-A: 5-5, Final Top-1 69.72%), which is indicated by the solid line. The comparison case indicated by the dotted line uses the same model and the same bit combination, but is without the selection tensor. Therefore, all quantization errors are included in the l2-norm regularization term. In other words, all quantization errors are removed uniformly during training process which is actually undesirable in our case.

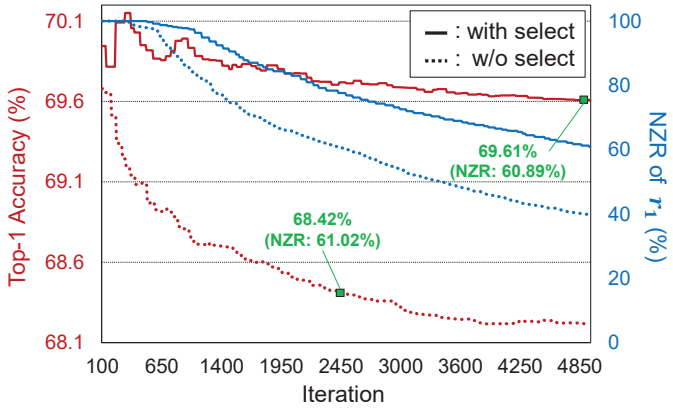

Figure 1. Effect of Select Tensor.

The above two cases have the same training options except for the selection tensor. NZR is a non-zero ratio of values in r_1 tensor and 1 epoch has 5005 iterations in total and we conducted 10 epochs as fine-tuning which described in the paper. We can identify the following features through the trends in Figure 1. First, the with-select case, which applies selection tensor, has a small decrease in accuracy. This high-accuracy trend is because, through the *select ratio*, some of the high-ranking quantization errors, which have a large impact of error compensation, are forcibly preserved. Furthermore, in early iterations, performance goes slightly above the baseline (69.75%). Second, the with-select case shows small decrease in NZR as well and the reason is due to the select tensor in training. By select tensor, some of scaled error indication values (15% in this case) in r_1 are excluded from l2-norm regularization term in task loss, so the corresponding penalty to r_1 is also reduced.

Therefore, for a fair comparison, if we pick the cases where similar NZR is reached in both two cases, it can be confirmed that the accuracy improved by about **1.2%** in the with-select case. Further after above training iterations, until 10 epochs,

there was a slight accuracy improvement without dramatic change, and finally, as we suggested in Table (4) from the paper, the with-select case converged to 69.72% and the without-select case to 68.51%. Of course, for the without-select case, there is the possibility of finding the optimal result by changing hyperparameters such as learning rate, weight decay, batch size, training scheduler and so on while observing the change in NZR. However, this manual search for the optimal point (even for sub-optimal points as a compromise) takes quite a long time especially for large models and datasets, and we also could not proceed with such an experiment due to a time problem.

In conclusion, our proposed method is an automation-based approach for quantization errors by initially selecting fixed, desired NZR, rather than finding numerous combinations and NZRs manually. As can be seen from the results presented in the paper, our method makes it possible to find a high-performance model automatically, even for a very low NZR such as 5%.

2. The Effect of Per-Tile Quantization (PTQ)

In the paper, we proposed the quantization of *weight tile* units for hardware optimization called PTQ. Since PTQ has a lower granularity than the element-level training method which we mainly propose, the accuracy can be degraded and PTQ is a performance-hardware trade-off which is an optional stage. We present the performance trend in Figure 2 the with-PTQ case for ResNet-18 which also recorded when training models in Table (4) with the same training options as above Figure 1. This data will help understand how the performance trend changes depending on the presence or absence of PTQ. The gray lines are the without-select case in Figure 1 and is for reference only.

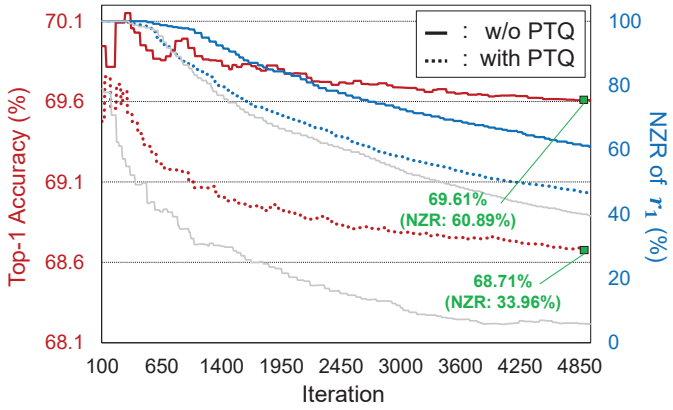

Figure 2. Effect of PTQ.

In the graph, we observe the following. First, with-PTQ case shows only a small degradation less than 1% in training accuracy compare to the without-PTQ case while performing additional hardware optimization. Second, because PTQ is a trade-off, users who need additional hardware efficiency can choose PTQ at the cost of very little accuracy degradation. The with-PTQ case indicated by the dotted line shows an accuracy drop of less than 1% (W-A: 5-5, Final Top-1 69.21%) compared to the without-PTQ case. This degradation trend is similar to the FLightNN [3] comparison cases presented in the paper. In conclusion, our proposed method provides an additional optimization for hardware deployment as well as a highly efficient log-scale quantization while incurring very little performance degradation.

3. Visualized Results

In the case of our applications, Image enhancement and Semantic segmentation, it would be helpful to visually present the actual image result compared to the full precision (FP) model for the quantized model that greatly reduced the number of bits. Therefore, image results for each application are attached as follows. For our models, we selected each model which has the lowest bit combination and lowest/various select ratio from Table (4) of the paper.

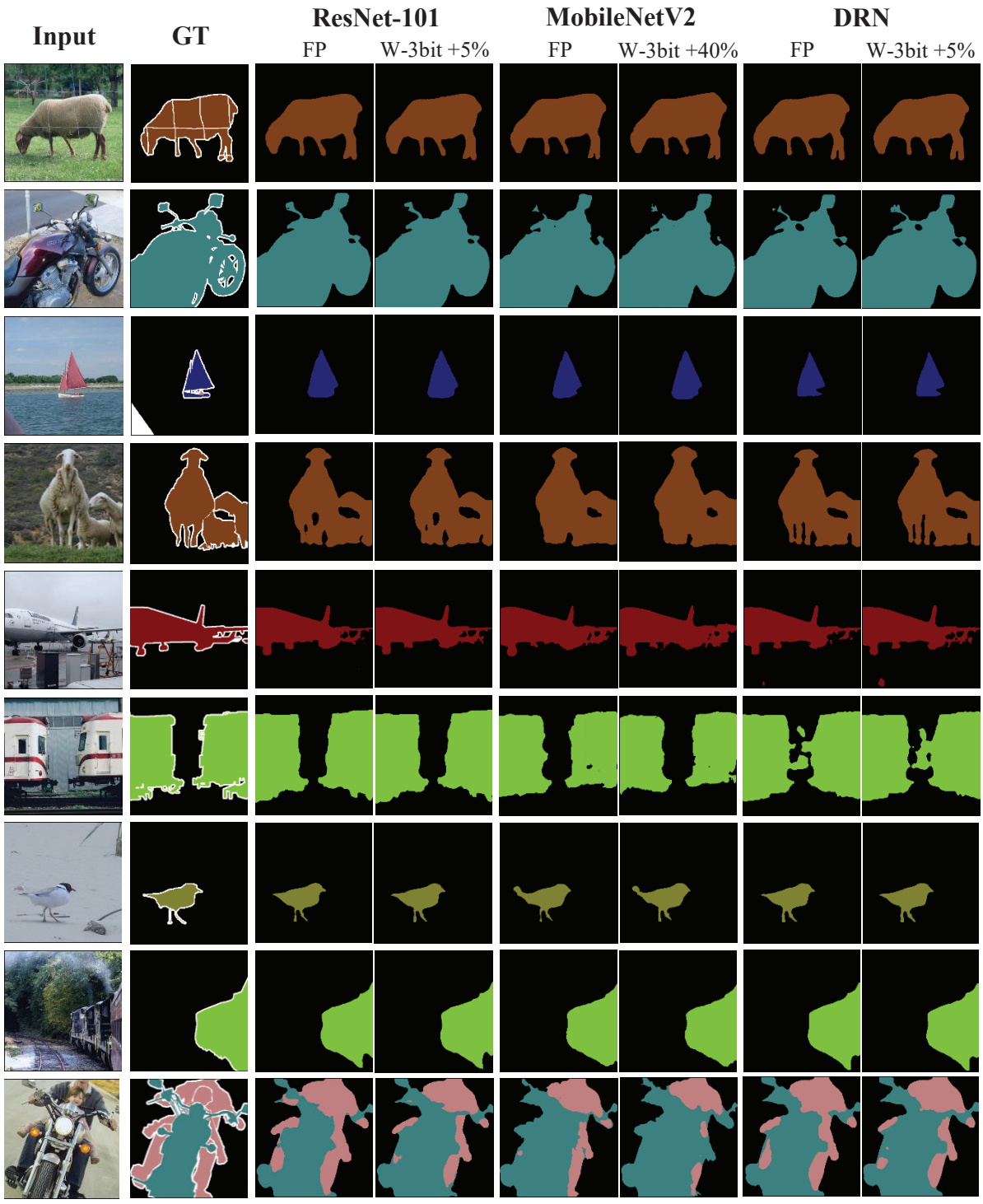

Figure 3. Visualization results with DeepLabV3+ [2] on PASCAL VOC 2012 [4] validation set. GT refers to ground truth and FP refers to full precision (32-bit), which is the baseline for each model. W-3bit + 15% means 3-bit of weight and 15% of select ratio. More detailed information is described in the paper.

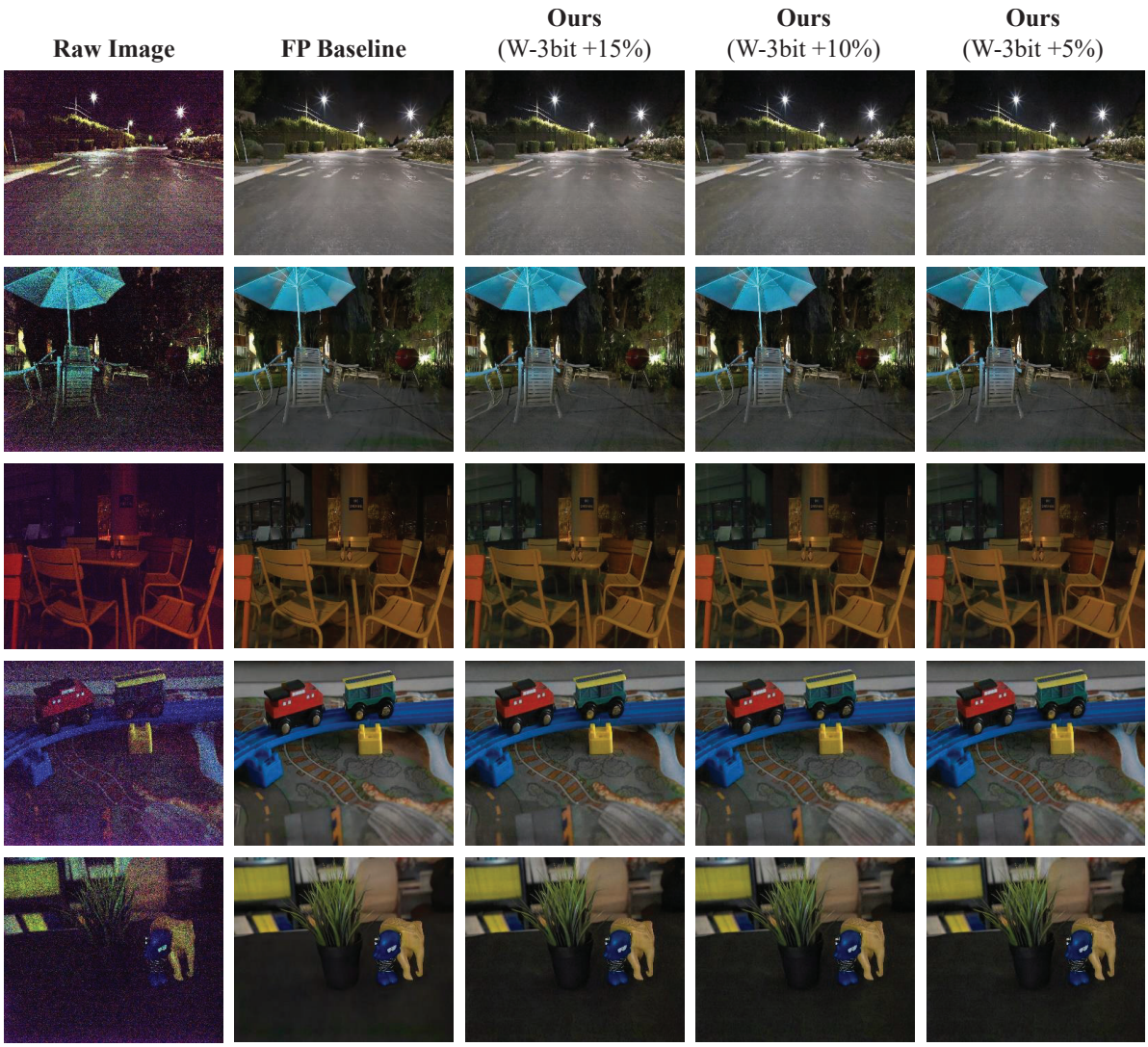

Figure 4. Visualization results with SID [1] on Sony validation set. FP refers to full precision (32-bit). W-3bit + 15% means 3-bit of weight and 15% of select ratio. More detailed information is described in the paper.

References

[1] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *Eur. Conf. Comput. Vis. (ECCV)*, 2018.

[3] Ruizhou Ding, Zeye Liu, Ting-Wu Chin, Diana Marculescu, and R. D. (Shawn) Blanton. Flightnns: Lightweight quantized deep neural networks for fast and accurate inference. *Proc. of the 56th Annual ACM/IEEE Design Automation Conference (DAC)*, 2019.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge (VOC)*, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.